# 中文信息处理－－基于特征提取的短信分级
# 实验报告

14307130318 刘超颖

## 一、研究背景

随着互联网的普及，在信息化高度发达的今天，人们的交流也变得愈加快捷和频繁，在移动通信领域，短信已经是我们日常生活的一部分，相比于其他通讯手段，短信具有普及性、实时性、异步性、可靠性和易于接近性等优势。作为全世界范围内最广泛使用的通讯媒介，短信记录了人们生活中的许多重要信息，比如重要通知、个人信息、具体数字，但也包含了大量无意义的对话，如拟声词、标点符号、表情符号等。尤其是在垃圾短信蔓延的当下，在短信通讯中，重要的短信消息往往被大量的垃圾短信淹没，这些垃圾信息被群发到所有用户的手机上，大多数是广告或宣传材料，如短期高利息贷款计划，私人商品贩售，赌博推广，色情网站推荐，诈骗信息等。

对短信智能管理的核心是能够对短信进行准确的分类，将所有短信按照重要程度进行分级，并将垃圾信息过滤掉。但是，对用户短信的分类仍困难重重，主要体现在以下几个方面：（1）没有大规模训练集：短信是私人数据，基本没有人愿意共享，所以获得大规模训练集是很难的一件事情。（2）文本太短：由于短信文本很短，这就导致特征值会多而散，非常不明显，很多的分类算法面对这种情况很难达到预想的效果。（3）短信息特有的奇异短语：很多字典中没有出现过的短语和缩写在日常生活和短信中被使用，对于这些短语，很难做出分词或者特征提取等操作。

本文将自然语言处理运用到手机短信分类研究中，通过分析短信的特点，综合用户习惯，合理运用近似和假设，详细研究适合短信分类的特征提取方法并使用朴素贝叶斯分类算法，试图得到一个高效、准确的手机短信分类算法，实现一种高效可行的短信管理方案。

## 二、基本思路

在日常生活中，我们收到的短信重要程度往往不一样，重要的如工作通知、时间安排，次重要的如与朋友的聊天记录，无意义的如群发节日祝福、验证码，和垃圾信息如广告、诈骗短信。故我们想通过一个自然语言处理系统，将这些短信进行分级，分为“重要短信”、“日常短信”、“无效短信”和“垃圾短信”四类（见图(1)），以便用户对这些短信进行批量操作和管理。

对于数据的选取，可以通过脚本爬取自己手机上的所有短信数据，也可以查找相关的语料库，如新加坡国立大学2004年收集的短信语料库(NUS SMS Corpus)。在本实验中，通过爬取自己手机中所有的短信，并加入网络上做过相关实验的开源短信资料，整合出message.xml文件。

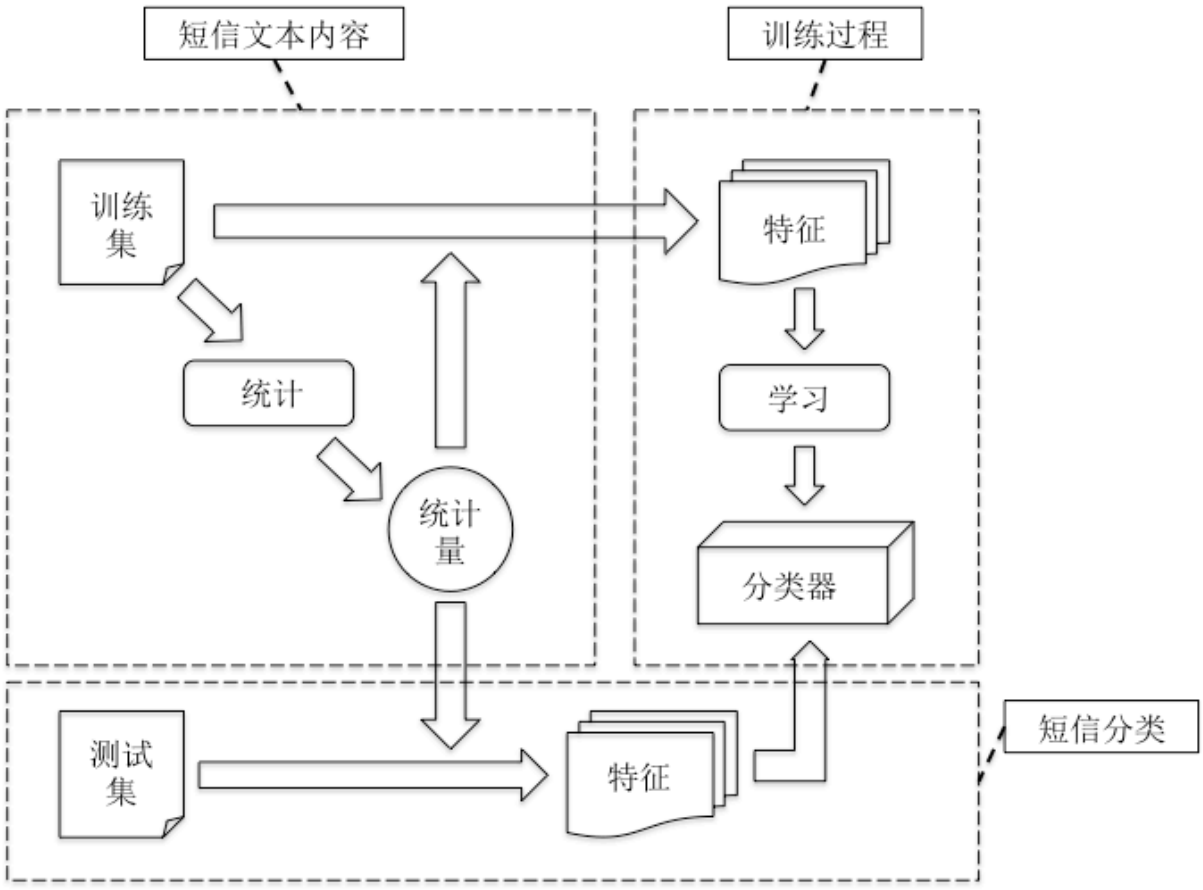
对于短信分级的基本实现流程，可用图(2)流程图表示。

其中，将已有的短信数据分为训练集和开发集，训练集用于提取特征和训练，而开发集作为开发时的测试集对训练结果进行测试，以评估算法的准确率和召回率。

对于短信内容的特征提取，最简单的想法是将训练集中的每条短信中的所有词汇存入一个列表，然后做频度分析，统计所有词语出现的频率，将每一级短信中出现频率最高的词汇和短语作为特征。在测试时，将这些特征与测试集中的文本内容进行比对，如果符合度较高，则将测试集中的短信分为特征相应级别的短信。


12-1233100
1069018800260
+66 6-4950-2062
【上海交警】您已成功预约科目三第26考点锦隆12月30日第一场的科目三 道路驾驶技能考试考试，请携带身份证明按时参加考试。如有特殊情况，请提前去车管所取消预约。具体场次签到时间请查看预约凭证。
大佬你在摸鱼吗？来啊快活啊～
【快的打车】5845（快的打车验证码）
【澳门最热门博彩公司推荐，注册即送高额彩金】
【菲律宾太阳城集团】免费注册：6661404.com 注册即送28元幸运彩金；MG老虎机、PT电子、百家乐、老虎机、六合彩、时时彩、顶级体育投注盘口，每周上千场赛事！无需转换额度，日结现金返水高达1.8%（可直接取款）
【澳门金沙集团】官网注册：5550349.com 注册获赠88幸运彩金；全球最大博彩公司，1元即可存取款，每月3号加赠彩金最高可达88888元，单笔取款最高1000万元；大额VIP玩家首选！
【澳门新葡京赌场】免费注册：0108706.com 注册获赠88幸运彩金；澳门四大财团重金打造亚洲最大线上真人真钱百家乐,电子游戏平台!
【澳门百家乐开户】免费注册：6661420.com 注册获赠158幸运彩金；专营百家乐、龙虎、轮盘、三
重要短信
日常短信
无效短信
垃圾短信

图(1)


短信文本内容
训练过程
训练集
特征
统计
学习
统计量
分类器
测试集
特征
短信分类

图(2)

每条短信存入数据集后格式如下：

```
<SMS>
    <Type>1</Type>
    <Status>0</Status>
    <Read>1</Read>
    <Protocal>0</Protocal>
    <Subject></Subject>
    <Address>106907301631</Address>
    <Date>2016-12-30 18:14:56</Date>
    <SERVICE_CENTER></SERVICE_CENTER>
    <Body><![CDATA[【网易】亲爱的阴阳师SAMA，寮办邀您重回平安京，《阴阳师》新年祭等你同乐！回归积分可兑换丰厚奖励，更有四大新式神重磅登场！姑获鸟特典皮肤盛装而来，新副本、新年福袋全面上线！快来看看 http://yys.163.com/m/mail 。回复TD退订]]></Body>
</SMS>
```

## 三、中文分词

由于中文语言自身的特点决定，在语言交流过程中，最基本的单位是“字”, 这一特点与英语等语种完全不同。就实际人与人之间的交流来看，仅仅凭借“字”这一概念会给交流带来歧义从而导致无法正常交流，同样地在文本分类的实际应用过程中，以“字”为单位进行特征抽取会必然会降低分类的效果。

目前在文本分类过程中，大多采用分词作为特征进行抽取。在中文信息处理领域，中文分词技术已有了一定的发展，较为流行的分词技术主要有 ICTCLAS 分词、正向最大匹配分词(FMM)、条件随机场分词(CRF)等,而本实验中将会用到已有分词工具——结巴中文分词，并在此基础上做一些优化和调整以达到更好的效果。在短信分级的实现中，以每条短信的文本内容为单位进行分词操作，最基本的单条短信的分词效果如下：

```
>>> seg_list = jieba.cut("【网易】亲爱的阴阳师SAMA，寮办邀您重回平安京，《阴阳师》新年祭等你同乐！回归积分可兑换丰厚奖励，更有四大新式神重磅登场！姑获鸟特典皮肤盛装而来，新副本、新年福袋全面上线！快来看看 http://yys.163.com/m/mail 。回复TD 退订", cut_all = False)
>>> print("/".join(seg_list))
【/网易/】/亲爱/的/阴阳师/SAMA/，/寮/办/邀/您/重回/平安/京/，/《/阴阳师/》/新年/祭/等/你/同乐/！/回归/积分/可兑换/丰厚/奖励/，/更/有/四大/新式/神/重磅/登场/！/姑获/鸟/特典/皮肤/盛装/而/来/，/新/副本/、/新年/福袋/全面/上线/！/快/来/看看/ /http/://///yys/./163/./com///m///mail/ /。/回复/TD/ /退订
```

## 三、特征提取

由于中文短信的数据集过小，大多数词语在短信中出现次数过少，短信分类面临着一个重要的问题，即特征维度太高，高维度的特征会在分类算法中带来很多问题。首先，维度过高会导致各维度之间的独立性变差，这会严重影响算法的准确率；其次，高维度的特征值会使算法效率下降，引入大量冗余的计算量；第三，高维度的特征值也会引入很多不必要的噪声。

所以，利用合理的特征值提取算法对特征空间降维是必要的，传统的文本特征值提取方法包括文档频率特征值提取方法和互信息特征值提取方法。由于互信息计算时完全没有考虑不同文档频率的词条对类别的判定能力的差异，所以当处理短信这种词条分散而训练集比

较小的数据时，将会出现很多文档频率只有1的词条，无法代表一个类别，这种方法显然是不适用的。故本实验将着重考虑词条文档频率特征值提取方法，并在这一方法的基础上进一步做一些处理，以实现短信的分类。

词条的文档频率(Document Frequency)是指在训练集中出现该词条的文档数。基于文档频率的特征值提取方法基于一个假设：当一个词条的文档频率小于某个阈值时，它并不具备或者很少具备类别区分的能力，需要将其作为噪声词去掉，从而达到降维的目的。

```
class MessageCountVectorizer(sklearn.feature_extraction.text.Count
Vectorizer):
    def build_analyzer(self):
        def analyzer(doc):
            words = pseg.cut(doc)
            new_doc = ''.join(w.word for w in words if w.flag !
= 'x')
            words = jieba.cut(new_doc)
            return words
        return analyzer
vec_count = MessageCountVectorizer(min_df = 2, max_df = 0.99)
data_count = vec_count.fit_transform(content)
```

以上代码中，在分词后，按比例删除df超过max_df或者df小于min_df的分词特征，以达到对特征空间降维的目的。在参数min_df取2，max_df取0.99的情况下，只要出现频数大于等于2次的分词都作为特征值，共产生41595个分词特征。

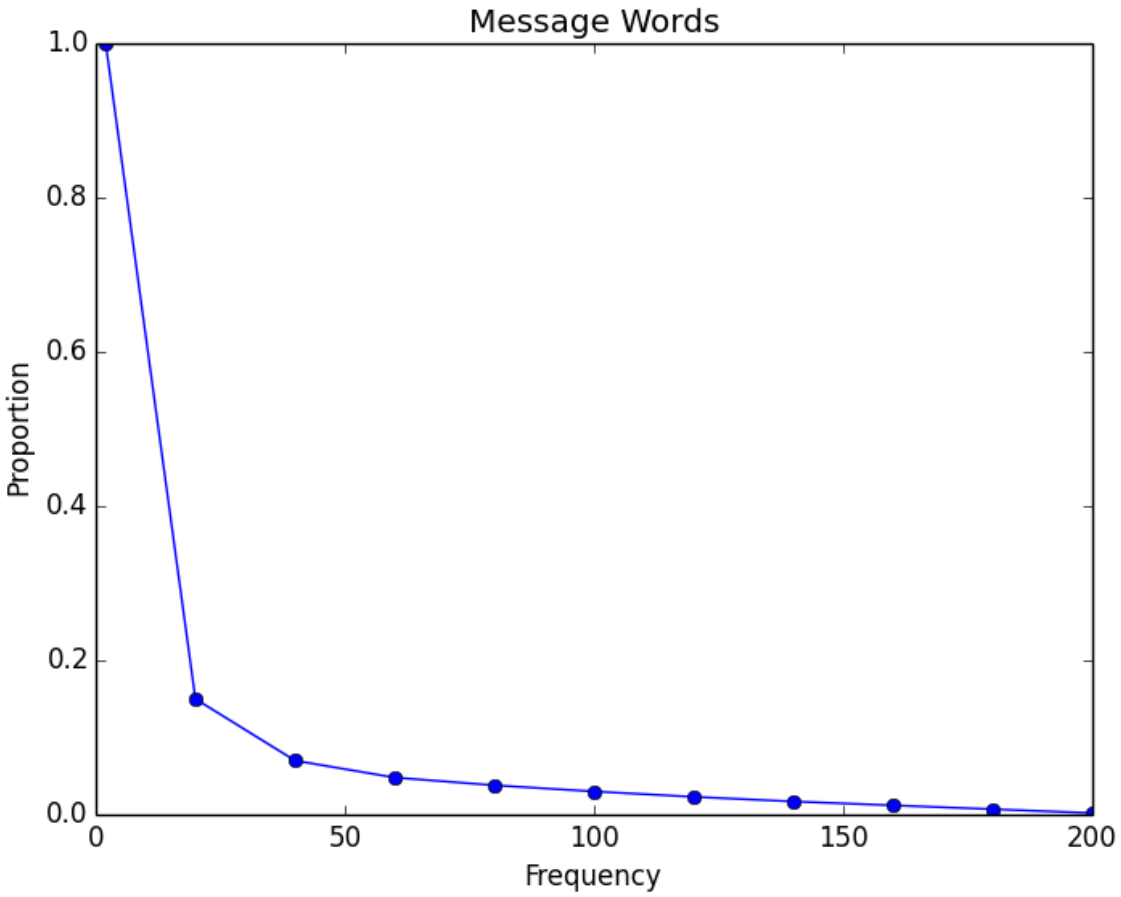
**四、频度分析**

在上述特征空间降维德过程中，我们引入了max_df和min_df两个参数，以筛选出出现频率满足一定条件的特征分词，但是我们仍面临着一个问题——如何取这两个参数才能使得特征筛选的结果达到较优?

在这种情况下，我们首先对分词出现的词频进行观察。图(3)反映单词在所有短信中的出现次数，即特征出现次数与出现次数大于等于该次的特征数目占总特征个数的比例，出现次数大于等于2次的分词的比例为1，随着次数增加比例依次减少。横坐标表示一个分词在训练集中出现的次数，纵坐标表示大于等于这一出现次数对应的分词个数在所有分词中的比例。从图中不难看到出现次数超过40次的特征仅占特征总数的10%不到。

根据图(3)反映的特征频率的性质，我们可以看出，对df_max的调整是无意义的，因为在频数高于一定次数后，函数图像无限趋近于0，而在频数较高的时候，函数的导数趋于0，图线的下降幅度极小，故我们将df_max的值固定为0.8，使得分词的df小于这一比例的词语能在一定程度上反应短信级别的特征。而对df_min的调整却是极为重要的，因为我们需要在出现多少次以上才能作为一个有效特征而非噪音和取足够精确的特征以确保算法的准确性这两个问题之间做出一个平衡。

取min_df分别为2，10，20，50时，使用朴素贝叶斯分类器随机测试10次得到短信分级的召回率，结果如表(1)、(2)、(3)、(4)。




图(3)

min_df=2时，特征分词有41595个:

| min_df=2 | 日常短信 | 垃圾短信 | 无效短信 | 重要短信 |
|---|---|---|---|---|
| max | 0.9696 | 0.9362 | 0.7647 | 0.9245 |
| min | 0.9531 | 0.7619 | 0.5789 | 0.8557 |
| mean | 0.9624 | 0.8616 | 0.6878 | 0.9004 |

表(1)

min_df=10时，特征分词有23917个:

| min_df=10 | 日常短信 | 垃圾短信 | 无效短信 | 重要短信 |
|---|---|---|---|---|
| max | 0.9729 | 0.9362 | 0.8095 | 0.9245 |
| min | 0.9440 | 0.7778 | 0.5789 | 0.8458 |
| mean | 0.9612 | 0.8648 | 0.6938 | 0.8993 |

表(2)

min_df=20时，特征分词有6239个：

| min_df=20 | 日常短信 | 垃圾短信 | 无效短信 | 重要短信 |
|---|---|---|---|---|
| max | 0.9615 | 0.9149 | 0.7647 | 0.9069 |
| min | 0.9449 | 0.7302 | 0.5455 | 0.8159 |
| mean | 0.9567 | 0.8097 | 0.6537 | 0.8789 |

表(3)

min_df=50时，特征分词有2454个：

| min_df=50 | 日常短信 | 垃圾短信 | 无效短信 | 重要短信 |
|---|---|---|---|---|
| max | 0.9574 | 0.8333 | 0.8095 | 0.9096 |
| min | 0.9351 | 0.7018 | 0.5263 | 0.7662 |
| mean | 0.9467 | 0.7754 | 0.6718 | 0.8510 |

表(4)

从结果可以看到特征越多，召回率越高，这点较为符合直觉。在短信分级的结果中，可以看出日常短信识别率较高，重要短信识别率居中，而对垃圾短信和无效短信识别的召回率尤其低下。

## 五、TF-IDF统计方法

关于召回率不高的原因，首先考虑的是特征选取方面。之前实验的分词特征的选取都是简单地将每一级短信中出现频数最高的词语提取出来作为特征值。而事实上，我们可以想象，在日常生活中任意两级短信，都有可能会出现一些频率在这两级短信中都非常高的分词。比如：在重要短信和垃圾短信中，都极有可能出现“邀请”，“支付”等分词；在无效短信和垃圾短信中，都极有可能出现“注册”，“操作”等分词。

对于这一问题，我们试图引入TF-IDF方法来解决。TF-IDF是一种统计方法，用以评估一字词对于一个文件集或一个语料库中的其中一份文件的重要程度。字词的重要性随着它在文件中出现的次数成正比增加，但同时会随着它在语料库中出现的频率成反比下降。

这一方法的主要思想是：如果某个词或短语在一篇文章中出现的频率TF高，并且在其他文章中很少出现，则认为此词或者短语具有很好的类别区分能力，适合用来分类。TF(Term Frequency)表示词条在文档d中出现的频率。IDF(Inverse Document Frequency)的主要思想是：如果包含词条t的文档越少，也就是n越小，IDF越大，则说明词条t具有很好的类别区分能力。

设想将TF-IDF方法引入短信分级后，如果一个分词在一级短信中频繁出现，则说明该分词能够很好代表这一级短信的特征，给这样的分词赋予较高的权重，并选来作为该级短信的特征词以区别与其它级的短信。

使用TF-IDF方法生成每一级短信的特征代码如下：

```
class MessageTfidfVectorizer(sklearn.feature_extraction.text.Tfidf
Vectorizer):
    def build_analyzer(self):
        #analyzer = super(TfidfVectorizer, self).build_analyzer()
        def analyzer(doc):
            words = pseg.cut(doc)
            new_doc = ''.join(w.word for w in words if w.flag !
= 'x')
            words = jieba.cut(new_doc)
            return words
        return analyzer
vec_tfidf = MessageTfidfVectorizer(min_df=2,max_df=0.8)
data_tfidf = vec.fit_transform(content)
```

min_df取2，随机测试10次得到短信分级的召回率如表(5)：

| | 日常短信 | 垃圾短信 | 无效短信 | 重要短信 |
|---|---|---|---|---|
| max | 0.9779 | 0.8867 | 0.7333 | 0.9663 |
| min | 0.9626 | 0.8371 | 0.4545 | 0.9496 |
| mean | 0.9713 | 0.8619 | 0.5866 | 0.9569 |

表(5)

由表(5)的数据可以看出，日常短信、垃圾短信和重要短信的识别率都有一定提升，但对无效短信识别的召回率反而有下降。

究其原因，我们发现对于验证码类的无效短信的识别非常准确，而对于另一种无效短信，如运营商发来的功能提示“【iPhone用户专享新功能！恭喜您获得iPhone语音信箱3个月免费试用期】当您手机无法接通时，“来电提醒+语音留言”让您不再错过重要来电。有创意的你还能自录一段应答语，如：“朕日理万机，尔等有事留言，无事退朝！”让您的iPhone酷起来！3个月到期后将自动结束试用。回复TDTY取消免费试用。详情请戳http://yyxx.10086.cn/m 或咨询10086”，其本身特有的特征不明显，极易被归入垃圾短信一类；或办理运营商业务时发送的短信“10086”，短信本身只包含极少量的数字，无法正确分级。

## 六、奇异短语识别

近几年，随着互联网的发展，更多的网络词汇和奇异短语不断出现，给自然语言处理技术带来挑战。而短信中也愈发多地包含了这类网络语言，一般这些语言的内容信息杂乱无章，传统自然语言处理技术对其中的一些特殊用法束手无策，例如“蓝瘦，香菇！本来今颠高高兴兴，泥为什莫要说这种话？”(难受，想哭！本来今天高高兴兴，你为什么要说这种话？)，通过已有分词工具处理后的分词结果是“蓝瘦/，/香菇/！/本来/今颠/高高兴兴/，/泥为/什莫/要说/这种/话/？”。又如上文中对《阴阳师》广告的分词，其中对“寮办”、“新年祭”、“式神”、“姑获鸟”等词语的分词都不准确，而这些分词方面的不准确性可能会导致短信分级的错误。

通过对奇异短语的观察，我们发现奇异短语分为两种，一种是在特定语境下的新型的名词，如“姑获鸟”、“寮办”等；另一种遵循一个基本规则，即语音映射，例如“蓝瘦”对应着“难受”，“香菇”对应着“想哭”，都是通过方言语音映射得到的。由此可以总结出若干特征，在此基础上提出识别奇异短语的任务，通过手工标注形成奇异短语语料库，提取奇异短语词典和奇异短语特征集。通过自行载入奇异短语词典，使得已有分词系统能识别奇异短语。

在引入奇异短语识别后，再次对短信的召回率进行10次测试，其结果如下表：

| | 日常短信 | 垃圾短信 | 无效短信 | 重要短信 |
|---|---|---|---|---|
| max | 0.9796 | 0.8756 | 0.8435 | 0.9713 |
| min | 0.9663 | 0.8417 | 0.5126 | 0.9395 |
| mean | 0.9729 | 0.8594 | 0.6243 | 0.9527 |

表(6)

通过表(6)的结果，可以看出，对奇异短语的分词结果进行处理后，短信的分级准确率并未有明显的提升或下降，分析其原因，可能有以下三个原因：

（1）奇异短语出现的范围小，往往出现在“日常短信”这一分级中，而极少出现在“重要短信”中，而在未引入奇异短语识别前，对“日常短信”这一级别的判定已经非常准确，加入奇异短语识别后不能再对其准确率做进一步的大幅度提升。

（2）奇异短语的出现频率小，由于训练集过小，这些奇异短语极有可能只在训练集中出现过一次，故在选取特征时极有可能将这些奇异短语筛选出去，从而使奇异短语不作为一类短信的特征，不影响短信分级的结果。

（3）奇异短语的分词结果与原本结果相差小，即使机器并未真正地正确理解奇异短语的意义，但极有可能分词的结果原本就与其真正含义的分词结果相似。如“蓝瘦，香菇！”的分词结果本身就是“蓝瘦/，/香菇/！”，与原本“难受，想哭！”的分词结果“难受/，/想哭/！”相同，在这种情况下，奇异短语未能对分词造成任何影响。

## 七、特征优化

（1）过滤停用词

直观认为，停用词、标点符号等信息由于没有实际的意义，在文本分类过程中并不能帮助提高文本分类的准确度。本实验所使用的Bigram停用信息主要包括“如果”、“可是”、“怎么”等信息。通过滤除Bigram停用词信息，本实验所用的中文短信数据集中共减少Bigram停用词信息497个；而滤除标点符号后，本实验所用的中文短信数据集共减少信息量9624个分词。

```
def process_stopwords(content):
    stopword = loaddata.load_stopword()
    content_after_stop = []
    for i in content:
        if i not in stopword:
            content_after_stop.append(i)
    return content_after_stop
```

（2）给电话号码创建新的特征

在短信中经常出现电话号码这样的一长串数字的信息，如果不对其进行处理，几乎所有的电话号码信息都不可能作为特征用来判断一类信息。而事实上，电话号码的出现在很大程度上反映了一些信息，直观上可以想象，如果一条短信中出现了电话号码，则该条短信极有可能是“重要短信”或“垃圾短信”。

对于这个问题，引入的处理方法是：创建一个新的特征，其定义为数字的个数。即当连续收到一长串数字后，判断数字串的长度，长度相同的数字串定义为同一特征。采用正则表达式，取出连续的数字，依次按照其维度分别赋值，共设置16个维度，1~15以及15以上。对于日常短信来说，许多短信是含有电话号码之类的，如果只以实际的电话作为维度，则遇到不同的电话号码，不能识别，因此设置该维度。

```
def process_cont_numbers(content):
    digits_features = np.zeros((len(content), 16))
    import re
    for i, line in enumerate(content):
        for digits in re.findall(r'\d+', line):
            length = len(digits)
            if 0 < length <= 15:
                digits_features[i, length-1] += 1
            elif length > 15:
                digits_features[i, 15] += 1
    return digits_features
```

（3）给电子邮箱地址创建新的特征

在短信中经常出现电子邮箱地址这样的信息，与电话号码类似的是，出现电子邮箱地址的短信也往往被分级为“重要短信”或“垃圾短信”。用正则表达式

```
([a-zA-Z0-9_\.\-])+\@(([a-zA-Z0-9\-])+\.)+([a-zA-Z0-9]{2,4})+
```

判定一段字母数字字符串是否是邮箱，然后创建一个新的特征，用于表示一条短信是否包含邮箱形式的字符串。

```
def process_cont_email(content):
    ema_features = 0
    import re
    for i, line in enumerate(content):
        for digits in re.findall(r'([a-zA-Z0-9_\.\-])+\@(([a-zA-
Z0-9\-])+\.)+([a-zA-Z0-9]{2,4})+', line):
            ema_features += 1
    return ema_features
```

（4）繁体字

直观上想象，在现实生活中肯定有短信发送者喜欢使用繁体字，繁体字使用在分词时不会造成影响，但在特征提取时，有可能因繁简字体的不同而导致同一个分词被重复提取为特征。因此在进行处理的时候可以引入繁体字表，考虑在进行处理之前先进行繁体字的替换。

```
def process_fantizi(content):
    fantizi = loaddata.load_fantizi()
    content_after_fantizi = []
    processed = set()
    for i in content:
        new_words=''
        for k in i:
            if k in fantizi:
                new_words += fantizi[k]
                processed.add((k, fantizi[k]))
            else:
                new_words += k
        content_after_fantizi.append(new_words)
    return content_after_fantizi, processed
```

（5）拆分字

通过对数据集的观察分析，我们找到了不少拆分字，如“月半”、“口可”、“禾兑”等都是在短信中常出现的拆分字。因此在进行处理的时候可以引入拆分字表，考虑在进行处理之前先进行拆分字的替换。

```
def process_chaifenzi(content):
    chaifenzi = loaddata.load_chaifenzi()
    content_after_chaifenzi = []
    found_chaifenzi = set()
    for line in content:
        result = line
        for k,v in chaifenzi.items():
            if k in line:
                found_chaifenzi.add((k,v))
                result = result.replace(k,v)
        content_after_chaifenzi.append(result)
    return content_after_chaifenzi, found_chaifenzi
```

在引入上述所有特征优化后，再次对短信的召回率进行10次测试，直接取精确度和召回率的平均值，其结果如下表：

| | 日常短信 | 垃圾短信 | 无效短信 | 重要短信 |
|---|---|---|---|---|
| precision | 0.9714 | 0.8953 | 0.8095 | 0.9665 |
| recall | 0.9541 | 0.9004 | 0.7778 | 0.9879 |

表(7)

## 八、结果分析

观察表(8)，发现实验对日常短信和重要短信的分级识别已经非常完善，而对垃圾短信和无效短信的分级识别准确度仍然不高。其原因在于：（1）训练集过小，样本空间不够导致训练结果不准确；（2）各级短信的特征往往具有相似性，在短信级别间有模糊边缘，有时人工标注时也无法精确判别某条短信属于哪一级；（3）部分短信内容由少量纯数字组成，很难通过特征提取的方法对其进行分级。

事实上，在生活中，我们往往只需要区分出日常短信和重要短信，对于垃圾短信和无效短信，我们并不关心其具体的分级，尤其是在批量删除信息时，甚至可以将其归为一类统一删除。但是，如果我们想要设置类似于“拒收垃圾短信”的功能，对于无效短信和垃圾短信的界定还是有必要的，在这一点上，本实验仍待改进。

## 九、参考资料

1. https://github.com/atupal/nlp/tree/master/split
2. https://github.com/fxsjy/jieba/
3. http://xh.5156edu.com/page/z2354m9952j19804.html
4. 杨柳, 殷钊, 滕建斌, 王衡, 汪国平. 改进贝叶斯分类的智能短信分类方法. 计算机科学. 第41卷第10期
5. Rohit Giyanani, Mukti Desai. Spam Detection using Natural Language Processing. IOSR Journal of Computer Engineering (IOSR-JCE)11